BI-ML1.21 přednáška 4 handout 16. října 2024
Problémy, návrhy apod. hlaste v GitLabu.
Verze souboru: 16. října 2024 07:40.

Co bude v dnešní přednášce

- Evaluace modelů obecně
- Připomenutí trénování, validace a testování
- Křížová validace
- Evaluace regrese
- Evaluace klasifikace

14 Evaluace modelů

Úvod do evaluace modelů

- Jednou z hlavních výzev strojového učení je, aby natrénovaný model dokázal dobře fungovat i na nových vstupech, které doposud neviděl.
- Této schopnosti říkáme schopnost *generalizace*.
- V typickém scénáři máme několik kandidátů na finální model a potřebujeme mezi nimi vybrat ten nejlepší.
- K tomuto je třeba mít nějakou kvantitativní míru výkonnosti modelu.
- Zaměřme se na problematiku měření výkonnosti modelů v rámci supervizovaného učení.
- V tomto případě se může zdá volba míry přímočará (přesnost pro klasifikaci, MSE pro regresi), ale ve skutečnosti to není tak jednoduché a je třeba správně zvolit metriku, která odpovídá žádoucímu chování modelu.
- Například u regresní úlohy můžeme chtít preferovat model, který nedělá velké chyby, ale poměrně často dělá chyby menší. Anebo můžeme preferovat model, který vetšinou nedělá ani malé chyby, ale občas udělá hodně velkou.

Ztrátová funkce

- Uvažujme vysvětlovanou proměnnou Y a vektor příznaků (vstupů) $\boldsymbol{X} = (X_1, \ldots, X_p)^T$.
- Model predikuje $\hat{Y} \equiv \hat{Y}(\boldsymbol{X})$, které je funkcí $\boldsymbol{X}$.
- Chybu predikce Y pomocí $\hat{Y}$ obecně měříme pomocí tzv. *ztrátové funkce* (angl. *loss function*) L.
- Ztrátová funkce vhodným způsobem měří, jak dobře daný model predikuje konkrétní hodnotu.
- Typickou volbou v případě regresní úlohy je kvadratická ztrátová funkce měřící *kvadratickou chybu* (angl. *squared error*)
$$L(Y, \hat{Y}) = (Y - \hat{Y})^2$$
nebo L_1 ztrátová funkce měřící *absolutní chybu* (angl. *absolute error*)
$$L(Y, \hat{Y}) = |Y - \hat{Y}|.$$
- Kvadratickou chybu využívá třeba lineární regrese.

Ztrátová funkce binární klasifikace

- U binární klasifikace model často odhaduje pravděpodobnost

$$\hat{p} = \hat{\mathrm{P}}(Y = 1 | \boldsymbol{X} = \boldsymbol{x}).$$

- Poznamenejme, že to umí například rozhodovací strom, který může pro daný list, do kterého padne $\boldsymbol{x}$, vrátit $\hat{p}$ jako relativní počet reprezentantů třídy 1 z trénovací množiny v tom listu.
- Finální predikce vysvětlované proměnné je potom

$$\hat{Y} = \begin{cases} 1 & \text{když } \hat{p} > 1/2, \\ 0 & \text{jinak.} \end{cases}$$

- Typická ztrátová funkce, která se v takovém případě používá je (angl. *binary cross-entropy loss*)

$$L(Y, \hat{p}) = -Y \log \hat{p} - (1 - Y) \log(1 - \hat{p}).$$

- To reálně znamená

$$L(1, \hat{p}) = -\log \hat{p} \quad \text{a} \quad L(0, \hat{p}) = -\log(1 - \hat{p}).$$

- Jak uvidíme později v semestru, tato ztrátová funkce odpovídá tomu, co se děje u logistické regrese.

Trénovací chyba

- Při učení se snažíme minimalizovat chybu predikce měřenou pomocí průměrné hodnoty ztrátové funkce L na trénovacích datech

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} L(Y_i, \hat{Y}(\boldsymbol{x}_i)),$$

  kde trénovací množina je N dvojic $(Y_i, \boldsymbol{x}_i)$.
- Tato průměrná hodnota se nazývá *trénovací chyba* (angl. *training error*) a občas se značí jako $\overline{\mathrm{err}}_{\mathrm{train}}$. Někdy se jí také říká trénovací ztráta (angl. *training loss*).
- U regresní úlohy, použití kvadratické ztrátové funkce vede k minimalizaci *střední kvadratické chyby* (angl. *mean squared error*)

$$\mathrm{MSE}_{\mathrm{train}} = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2,$$

  což je ekvivalentní minimalizaci RSS v lineární regresi.

Trénovací chyba pokračování

- Použití L_1 ztrátové funkce vede na minimalizaci *střední absolutní chyby* (angl. *mean absolute error*)

$$\mathrm{MAE}_{\mathrm{train}} = \frac{1}{N} \sum_{i=1}^{N} |Y_i - \hat{Y}_i|.$$

- U binární klasifikace, použití předchozí ztrátové funkce vede k minimalizaci binární relativní entropie (angl. *binary cross-entropy*)

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \Big[Y_i \log \hat{p}(\boldsymbol{x}_i) + (1 - Y_i) \log(1 - \hat{p}(\boldsymbol{x}_i)) \Big].$$

- Časem uvidíme, že až na mínus před sumou se jedná přesně o logaritmus věrohodnostní funkce, který budeme maximalizovat u logistické regrese. S mínusem a minimalizací se tedy jedná o totožnou úlohu.

Učení modelu

- Během procesu učení modelu se zafixovanými hyperparametry se snažíme najít hodnoty parametrů, které minimalizují trénovací chybu.
- U některých modelů, jako např. lineární regrese, lze toto řešení nají explicitně.
- Pro většinu modelů to ale nelze a používají se různé iterativní metody (typicky založené na gradientním sestupu), které se snaží nalézt lokální minimum.
- Jakmile je model natrénován, zajímá nás jeho schopnost generalizace.
- Podívejme se nejprve na odhadování výkonnosti modelu na nových datech pomocí ztrátové funkce. Pozdeji si ukážeme i jiné míry měřící výkonnost, resp. schopnost generalizace.

Testovací chyba

- *Testovací chyba* (angl. *test error*), také někdy nazývaná *generalizační chyba* (angl. *generalization error*), je definována jako střední hodnota chyby na novém vstupu $\boldsymbol{X}$ podmíněná danými trénovacími daty,

$$\mathrm{Err}_{\mathcal{D}} = \mathrm{E}\left(L\big(Y, \hat{Y}(\boldsymbol{X})\big)|\mathcal{D}\right),$$

  kde $\mathcal{D} = \big((Y_1, \boldsymbol{x}_1), \ldots, (Y_N, \boldsymbol{x}_N)\big)$ značí trénovací data.
- Testovací chybu můžeme odhadnout výběrovým průměrem

$$\overline{\mathrm{err}}_{\mathrm{test}} = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} L\big(Y_i, \hat{Y}(\boldsymbol{x}_i)\big),$$

  změřeným na testovací datech $\big((Y_1, \boldsymbol{x}_1), \ldots, (Y_{N_{\mathrm{test}}}, \boldsymbol{x}_{N_{\mathrm{test}}})\big)$, které byla získány nezávisle na trénovacích datech $\mathcal{D}$.
- Poznamenejme, že $\overline{\mathrm{err}}_{\mathrm{test}}$ podmíněno trénovacími daty je nestranný odhad $\mathrm{Err}_{\mathcal{D}}$, t.j.

$$\mathrm{E}\left(\overline{\mathrm{err}}_{\mathrm{test}}\,|\mathcal{D}\right) = \mathrm{Err}_{\mathcal{D}}\,.$$

- Nejobecnější mírou schopnosti modelu generalizovat je *očekávaná testovací chyba* nebo taky *očekávaná chyba predikce* (angl. *expected test error*, *expected prediction error*) definovaná jako střední hodnota testovací chyby vzhledem k náhodnému výběru trénovací množiny

$$\mathrm{Err} = \mathrm{E}\left(\mathrm{Err}_{\mathcal{D}}\right) = \mathrm{E}\,L\big(Y, \hat{Y}(\boldsymbol{X})\big).$$

15 Vyhodnocovací scénáře

Vyhodnocovací scénáře

Chceme-li natrénovat model a pak odhadnout jeho testovací chybu, musíme mít k dispozici trénovací data a nezávislá testovací data.

- Kromě toho v mnoha případech máme více kandidátů na finální model a potřebujeme z nich vybrat ten nejlepší.
- Typicky se jedná o situaci, kdy máme celou třídu modelů závisejících na nějaké sadě hyperparametrů a chceme najít jejich hodnoty tak, aby testovací chyba finálně vybraného modelu byla co nejmenší.
- Z pohledu evaluace tak máme dva různé úkoly:
  - Výběr modelu - odhadnout výkonnost různých modelů za účelem výběru nejlepšího.
  - Ohodnocení modelu - odhadnout testovací chybu finálního modelu.
- Protože výběr modelu vlastně spadá do procesu trénování (vybereme model, který se nejlépe přizpůsobí datům) *nesmíme* použít stejná data pro výběr finálního modelu i pro ohodnocení tohoto finálního modelu.

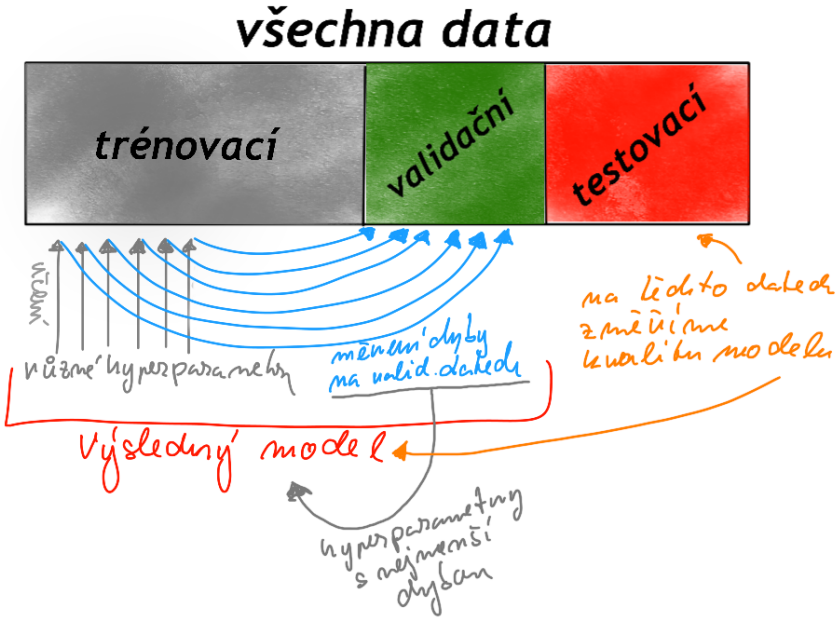
Trénování, validace a testování

Když máme *dostatek dat*, rozdělíme vstupní data na tři části: *trénovací* a *validační* a *testovací*.

- *Trénovací část* použijeme pro trénování konkrétních modelů se zafixovanými hyperparametry.
- *Validační část* použijeme k ohodnocení daného modelu a porovnání s ostatními modely. Takto vybereme nejlepší sady hyperparametrů a případně porovnáme nejlepší modely z různých tříd. Finální model tedy vybíráme na základě validační množiny.
- *Testovací část* použijeme až v závěrečné fázi, když už máme vybraný a natrénovaný finální model. Tím získáme odhad testovací chyby, kterou můžeme očekávat na nových datech.

Tento způsob práce s testovací množnou se nazývá *hold-out*.

Trénování, validace a testování - obrázek



Trénování, validace a testování - chyby

- V některých případech se objevuje nesprávné použití těchto tří částí datasetu.
- Validační část je (správně) použita k ladění hyperparametrů v rámci jedné třídy modelů.
- Potom jsou ale modely s nejlepšímy hyperparametry z různých tříd (např. KNN a rozhodovací strom) porovnávány (*špatně*) na základě testovacích dat!
- Protože *výběr nejlepšího modelu je součást trénování*, finální model je potom vybrán na základě všech tří částí datasetu.
- Tudíž je finální ohodnocení na testovacích datech příliš optimistické.
- Testovací množinu musíme vždy oddělit co nejdříve (ještě před předzpracováním) a držet ji striktně bokem pouze pro finální evaluaci.

Trénování, validace a testování - rizika

- Evaluace pomocí validačních nebo testovacích dat dává rozumné výsledky pouze pokud trénovací, validační a testovací data pochází ze stejného rozdělení a když tam nejsou žádné procesně vnesené odlišnosti (např. že by data byla nějakým způsobem uměle uspořádaná a před rozdělení by se neprovedla náhodná permutace).
- V mnoha aplikacích prediktivního modelování se systém, který sledujeme a chceme predikovat, v čase vyvíjí (je tzv. „nestacionární“). To může přinést systematické odlišnosti mezi trénovacími a reálnými daty v budoucnosti.
- Příkladem může být predikce cen na burze, kdy data za minulých 5 let typicky nebudou odpovídat současnosti.
- Pokud očekáváme takovéto chování, je dobré, aby testovací (a případně i validační) data správně reflektovala chronologické řazení a příslušné odhady chyb tak mohli reálně ukazovat tento posun v čase. V takovém případě tedy data před rozdělením typicky nepermutujeme.
- U vícekrokového modelování (kdy např. doplňujeme chybějící hodnoty nějakým modelem, vybíráme příznaky atd.) musíme validační a testovací data oddělit již na začátku a pak na ně aplikovat všechny tyto kroky „naučené“ na trénovacích datech.

16 Křížová validace

Křížová validace

- Když nemáme dostatek dat, nebývá rozumné dělit je na trénovací, validační a testovací část.
- Uvažujme nejprve scénář, kdy si testovací data dopředu oddělíme a jde nám pouze trénování a výběr nejlepšího modelu.
- V takovém případě můžeme k výběru nejlepšího modelu použít techniku *křížové validace* (angl. *cross-validation*).
- Základní metodou je tzv. k-násobná křížová validace (angl. *k-fold cross-validation*), kde $2 \leq k \leq N$:
  - Trénovací data $\mathcal{D}$ náhodně rozdělíme na k *podobně* velkých částí $\mathcal{D}_1, \ldots, \mathcal{D}_k$.
  - Pro každé $j = 1, \ldots, k$ model s danými hodnotami hyperparametrů natrénujeme na datech z množiny $\left(\cup_{i=1}^{k} \mathcal{D}_i\right) \setminus \mathcal{D}_j$.

Trénovací $\mathcal{D}_1$	Testovací $\mathcal{D}_2$	Trénovací $\mathcal{D}_3$	Trénovací $\mathcal{D}_4$	Trénovací $\mathcal{D}_5$

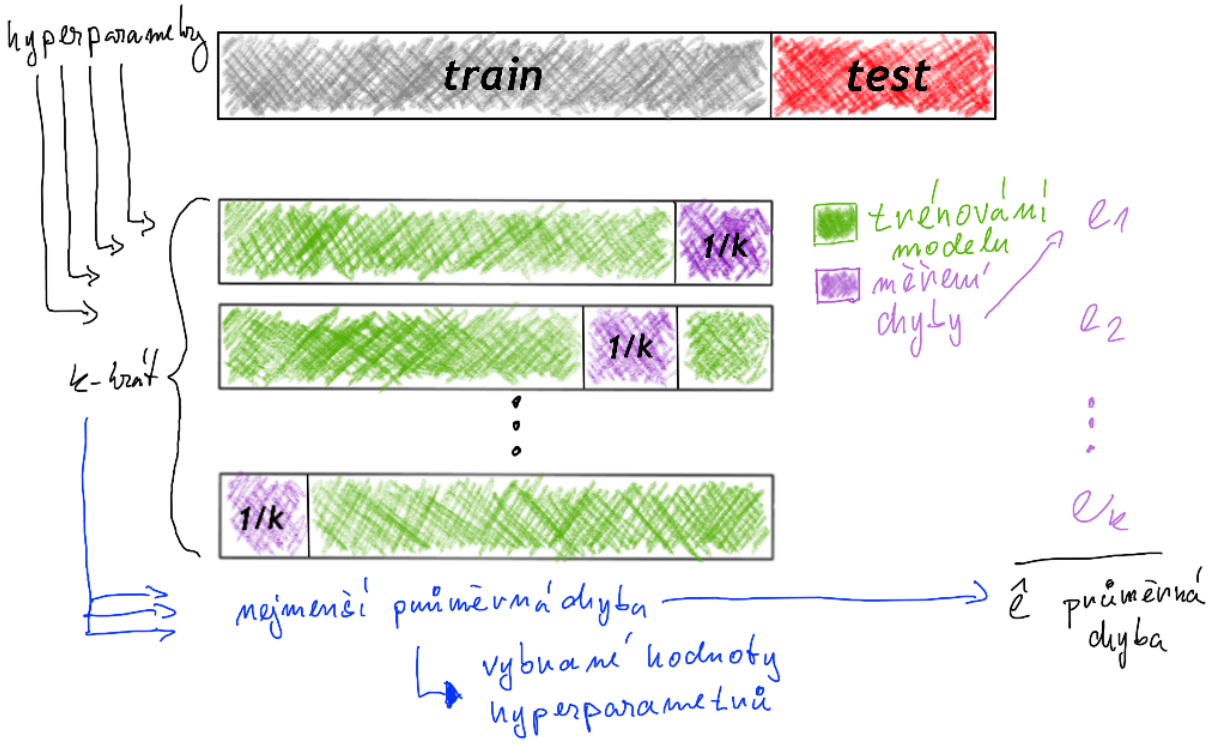
  - Na množině $\mathcal{D}_j$ odhadneme jeho chybu jako e_j.
  - Nakonec vrátíme průměrnou „cross-validační“ chybu

$$\hat{e} = \frac{1}{k} \sum_{i=1}^{k} e_i.$$

Křížová validace - pokračování

- Tento proces zopakujeme pro všechny zkoumané hodnoty hyperparametrů a na závěr vybereme jako nejlepší volbu ty hodnoty, které vedly k nejmenší cross-validační chybě.
- Abychom pak získali finální natrénovaný model (ten v tuhle chvíli nemáme), pro vybrané hodnoty hyperparametrů model znovu natrénujeme na celé trénovací množině $\mathcal{D}$!
- Typické volby k jsou 5 až 10.
- V extrémním případě, kdy se k rovná počtu trénovacích dat, mluvíme o **leave-one-out cross-validation**: trénuje se na celé trénovací množině bez jediného bodu, na kterém se pak měří výsledná chyba.

Křížová validace: jak to funguje (obrázek)



Křížová validace - diskuse

- Křížová validace může být neúnosně výpočetně náročná a nemůže tak vždy nahradit strategii s jedinou validační množinou.
- Pro fixní model je cross-validační chyba odhadem očekávané testovací chyby Err a nikoliv testovací chyby $\mathrm{Err}_{\mathcal{D}}$, což může být matoucí.
- Když máme opravdu málo dat a nechceme ani oddělovat testovací množinu, můžeme křížovou validaci použít dvoustupňově a současně tak vybrat nejlepší model i odhadnout jeho výkonnost pomocí očekávané testovací chyby. V tomto případě vnitřní křížovou validaci používáme na výběr nejlepšího modelu a vnější na odhad očekávané chyby.
- Výsledná vnější cross-validační chyba ale odpovídá očekávané chybě celé procedury pro výběr nejlepšího modelu, nikoliv chybě konkrétního modelu (který nám z toho ani nevypadne) a my tu proceduru pak musíme na celých datech provést a získat tak nejlepší model pro predikci (vizte [Cawley, Talbot (2010)]).

Strategie přípravy finálního modelu

- Při využití testovací množiny (hold-out přístup) získáme odhad testovací chyby nějakého nejlepšího modelu, který máme v tu chvíli i natrénovaný.
- U dvoustupňové křížové validace, získáme odhad očekávané testovací chyby procedury pro výběr nejlepšího modelu, ale nejlepší model teprve musíme získat.
- V tomto případě a i v tom prvním, pokud nemáme moc dat, má smysl sestrojit finální model úplně odznova a využít k tomu celá data.
  - Při hold-out přístupu s validační množinou tedy rozdělíme celý dataset pouze na 2 části - trénovací a validační. Různé modely (hyperparametry) pak trénujeme na trénovací části a pomocí validační části vybereme nejlepší model (hyperparametry). Dokonce tento model můžeme ještě znovu natrénovat na celých datech.

- Při hold-out přístupu s křížovou validací nebo při dvoustupňové křížové validaci vezmeme celý dataset a pomocí křížové validace vybereme nejlepší model (hyperparametry). Ten pak na celých datech natrénujeme.

- Ve žádném ze scénářů si neodložíme data pro měření výkonnosti finálního modelu.
- Jako odhad této výkonnosti použijeme odhad (očekávané) testovací chyby z předchozího kroku (hold-out s validační množinou, s křížovou validací nebo dvoustupňová křížová validace)

17 Evaluace regrese

Vyhodnocování regrese (1/2)

Nejobvyklejší volba kvadratické chyby jako ztrátové funkce vede na evaluaci pomocí *střední kvadratické chyby* (angl. *mean squared error*)

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2.$$

Tato míra penalizuje především velké odchylky a je velmi citlivá na odlehlé hodnoty.

Podívejme se na další vyhodnocovací míry, které se soustředí i na jiné aspekty predikce:

- *Root mean squared error* - odpovídá nelineárně přeškálovanému MSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}.$$

  Má stejné vlastnosti, akorát jednotky jsou stejné, jako jednotky vysvětlované proměnné.

Vyhodnocování regrese (2/2)

- *Root mean squared logarithmic error* - pro nezáporné hodnoty vysvětlované proměnné:

$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log Y_i - \log \hat{Y}_i)^2}$$

  Soustředí se na relativní míru odchylek - tj. pro malé hodnoty řeší i malé odchylky, pro velké hodnoty pouze ty velké. Je méně citlivá na odlehlé hodnoty.

- *Mean absolute error* - odchylky se skládají lineárně:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i|.$$

  Méně citlivá k odlehlým hodnotám než MSE.

- *Koeficient determinace* - R^2 (koeficient „R kvadrát“) vyjadřuje, jaký podíl variability cílové proměnné model vysvětluje:

$$R^2 = 1 - \frac{\mathrm{RSS}}{\mathrm{SST}},$$

  kde

$$\mathrm{RSS} = \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \quad \text{a} \quad \mathrm{SST} = \sum_{i=1}^{N}(Y_i - \bar{Y})^2.$$

  (RSS - residual sum of squares, SST - total sum of squares)

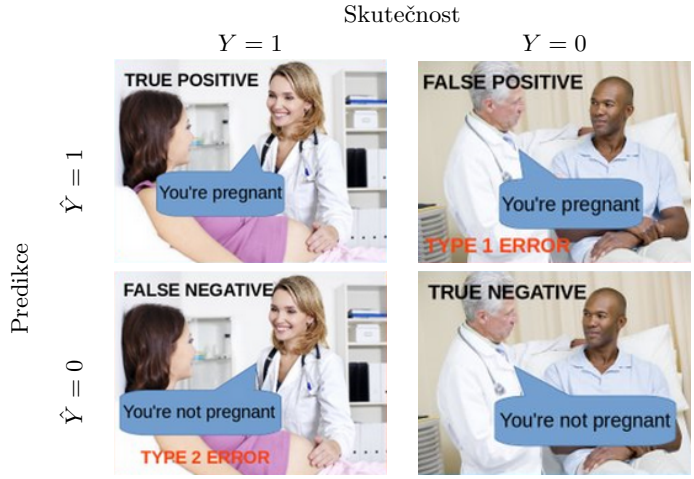
18 Evaluace klasifikace

Evaluace klasifikace - úvod

Při klasifikaci se chyba měřená pomocí ztrátové funkce pro rozumnou evaluaci příliš nehodí. Typicky se totiž jedná o relativní entropii, jejíž číselné hodnoty jsou těžko interpretovatelné.

Nejčastěji se vyhodnocování klasifikačních modelů provádí pomocí měr odvozených z tzv. *matice záměn* (angl. *confusion matrix*), což je matice četností různých predikovaných hodnot $\hat{Y}_i$ proti různým skutečným hodnotám Y_i.

Zaměřme se pouze na binární klasifikaci.



[zdroj: univ-angers.fr]

Matice záměn

Matice záměn spolu s řádkovými a sloupcovými součty:

		Skutečnost		
		$Y = 1$	$Y = 0$	$\sum$
Predikce	$\hat{Y} = 1$	TP	FP	$\hat{N}_+ = \mathrm{TP} + \mathrm{FP}$
	$\hat{Y} = 0$	FN	TN	$\hat{N}_- = \mathrm{FN} + \mathrm{TN}$
	$\sum$	$N_+ = \mathrm{TP} + \mathrm{FN}$	$N_- = \mathrm{FP} + \mathrm{TN}$	$N = \mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}$

V matici záměn jsou následující četnosti:

- *True positive* - TP - kolikrát model správně predikoval $Y = 1$.
- *False positive* - FP - kolikrát model *špatně* predikoval $Y = 0$.
- *False negative* - FN - kolikrát model *špatně* predikoval $Y = 1$.
- *True negative* - TN - kolikrát model správně predikoval $Y = 0$.

Řádkovými součty $\hat{N}_+$, resp. $\hat{N}_-$ jsou počty bodů, kde model predikoval 1, resp 0.
Sloupcovými součty N_+, resp. N_- jsou počty bodů ve třídě 1, resp 0.
Ideální predikce vede k tomu, že FN i FP jsou rovny 0.

Matice záměn

Z matice záměn můžeme odvodit následující míry, které odpovídají odhadům podmíněných pravděpodobností $\mathrm{P}(\hat{Y} = \hat{y} | Y = y)$:

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	$\mathrm{TPR} = \frac{\mathrm{TP}}{N_+}$	$\mathrm{FPR} = \frac{\mathrm{FP}}{N_-}$
$\hat{Y} = 0$	$\mathrm{FNR} = \frac{\mathrm{FN}}{N_+}$	$\mathrm{TNR} = \frac{\mathrm{TN}}{N_-}$

Pro tyto míry se používá několik různých označení:

- *True positive rate* (TPR) se také nazývá *sensitivita* nebo *recall* nebo *hit rate*.
- *False positive rate* (FPR) se také nazývá *false alarm rate* nebo *type I error rate*.
- *False negative rate* (FNR) se také nazývá *miss rate* nebo *type II error rate*.
- *True negative rate* (TNR) se také nazývá *specificita* nebo *selektivita*.

Kromě toho se někdy používají i odhady obrácených pravděpodobností $\mathrm{P}(Y = y | \hat{Y} = \hat{y})$.
Konkrétně především *precision* nebo také *positive predictive value*, což je odhad $\mathrm{P}(Y = 1 | \hat{Y} = 1)$:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\hat{N}_+}.$$

Nepoužívanější evaluační míry binární klasifikace

Podívejme se nyní na dvě důležité evaluační míry, které můžeme odvodit z matice záměn.

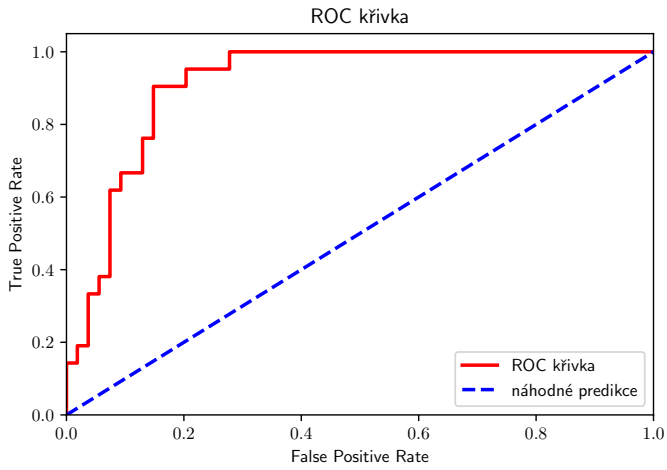
- *Přesnost* - odhad $\mathrm{P}(\hat{Y} = Y)$:
$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{N}.$$
Jedná se o suveréně nejpoužívanější míru.
- Přesnost není příliš vhodná, pro nevybalancované datasety, kde $\mathrm{P}(Y = 1)$ nebo $\mathrm{P}(Y = 0)$ je velmi malé. V takovém případě bude přesnost vysoká, jakmile model zvládne správně predikovat majoritní třídu.
- *F1 score* - harmonický průměr precision $\mathrm{P}(Y = 1|\hat{Y} = 1)$ a recall $\mathrm{P}(\hat{Y} = 1|Y = 1)$
$$F_1 = \frac{2}{1/\mathrm{PPV} + 1/\mathrm{TPR}} = 2\frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}}.$$
Tato míra je užitečná především pro nevybalancované datasety, kde $\mathrm{P}(Y = 1)$ je velmi malá.

Finální predikce v binární klasifikaci

- Podívejme se na úlohu binární klasifikace, kdy model v bodě $\boldsymbol{X}$ odhaduje pravděpodobnost $p(\boldsymbol{X}) = \mathrm{P}(Y = 1|\boldsymbol{X})$.
- Při klasickém přístupu (např. v logistické regresi) se predikce Y získá porovnáním této pravděpodobnosti s číslem 1/2
$$\hat{Y} = \mathbb{1}_{\hat{p}(\boldsymbol{X})>0.5}$$
tj. pokud je pravděpodobnost větší než 0.5 predikujeme třídu 1, v opačném případě predikujeme třídu 0.
- Tento způsob znamená, že predikujeme třídu s vyšší pravděpodobností.
- Ve skutečnosti ale můžeme zobecnit toto rozhodovací pravidlo zavedením nového hyperparametru $\tau \in [0, 1]$ a modifikací predikce tak, že
$$\hat{Y}_\tau \equiv \hat{Y}_\tau(\boldsymbol{X}) = \mathbb{1}_{\hat{p}(\boldsymbol{X})>\tau}.$$
- Platí tedy $\hat{Y} = \hat{Y}_{0.5}$.
- Pro každou hodnotu τ získáme trochu jiné predikce. Vždy ale platí, že pokud $\hat{Y}_\tau = 0$, potom určitě $\hat{Y}_{\tau'} = 0$ pro každé $\tau' > \tau$.
- V každém konkrétním bodě $\boldsymbol{X}$ tudíž platí, že predikce $\hat{Y}_\tau$ v závislosti na rostoucí hodnotě τ začíná na 1 pro $\tau = 0$ (kromě málo častého případu $\hat{p}(\boldsymbol{X}) = 0$) a poté v nějaké konkrétní hodnotě $\tau \in (0, 1)$ přeskočí do 0 a tam zůstane až do $\tau = 1$.

ROC křivka

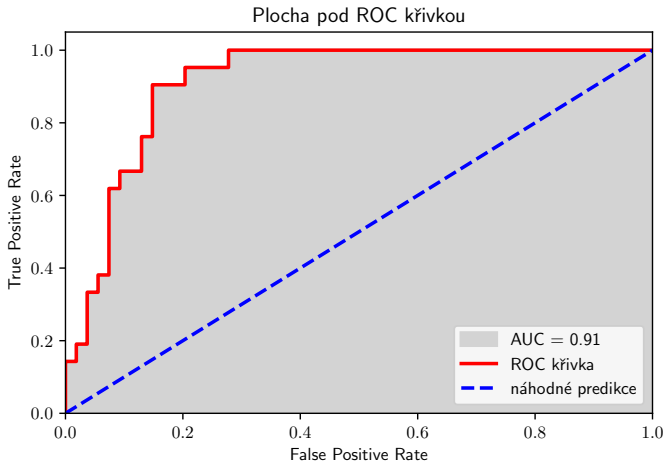
- Podívejme se nyní na chování true positive rate (TPR) a false positive rate (FPR) v závislosti na τ.
- Z předchozího slajdu plyne, že jsou obě neklesající funkce od τ.
- Graf TPR_τ versus FPR_τ jakožto implicitní funkce τ se nazývá *receiver operating characteristic* nebo také *ROC křivka*.



- Pro dobrý model, bude graf strmě stoupat k levému hornímu rohu a poté již jen velmi pomalu stoupat do pravého horního rohu, kde musí končit.
- Přímka na diagonále odpovídá náhodné predikci, kdy $\mathrm{TPR}\tau \doteq \mathrm{FPR}_\tau$ pro každé τ.

AUC - plocha pod ROC křivkou

- Kvalita modelu, pro který máme ROC křivku se obvykle vyhodnocuje jedním číslem, které znamená *plochu pod křivkou* (angl. *area under curve*) a značí se *AUC*.
- Model s náhodnými predikcemi bude mít plochu pod křivkou 0.5.
- AUC dokonalého modelu bude rovno 1.
- Obvykle má většina modelů AUC mezi 0.5 a 1.



ChangeLog

Verze	Datum	Autor	Log
1.0	16.10.2024	DV	Výchozí verze pro rok 2024/2025.
1.0	26.10.2023	DV	Výchozí verze pro rok 2023/2024.
1.0	22.11.2022	DV	Výchozí verze pro rok 2022/2023.